

**AN
INTERNSHIP REPORT
ON
GREENSCREEN BACKGROUND REMOVER USING CV
SYSTEM REPORT
BY
KAMAL ACHARYA
(Tribhuvan University)**

Date: 2022/06/07

Abstract

A green screen is a backdrop against which footage is shot to later be superimposed over a separately-filmed background in a visual effects process called “chroma keying.” It is also known as chroma key compositing, colorkeying, or simply keying..

This term describes removing the green screen background in post-production using video editing software. When the green screen background has been keyed, it'll be fully transparent..

TALBLE OF CONTENTS

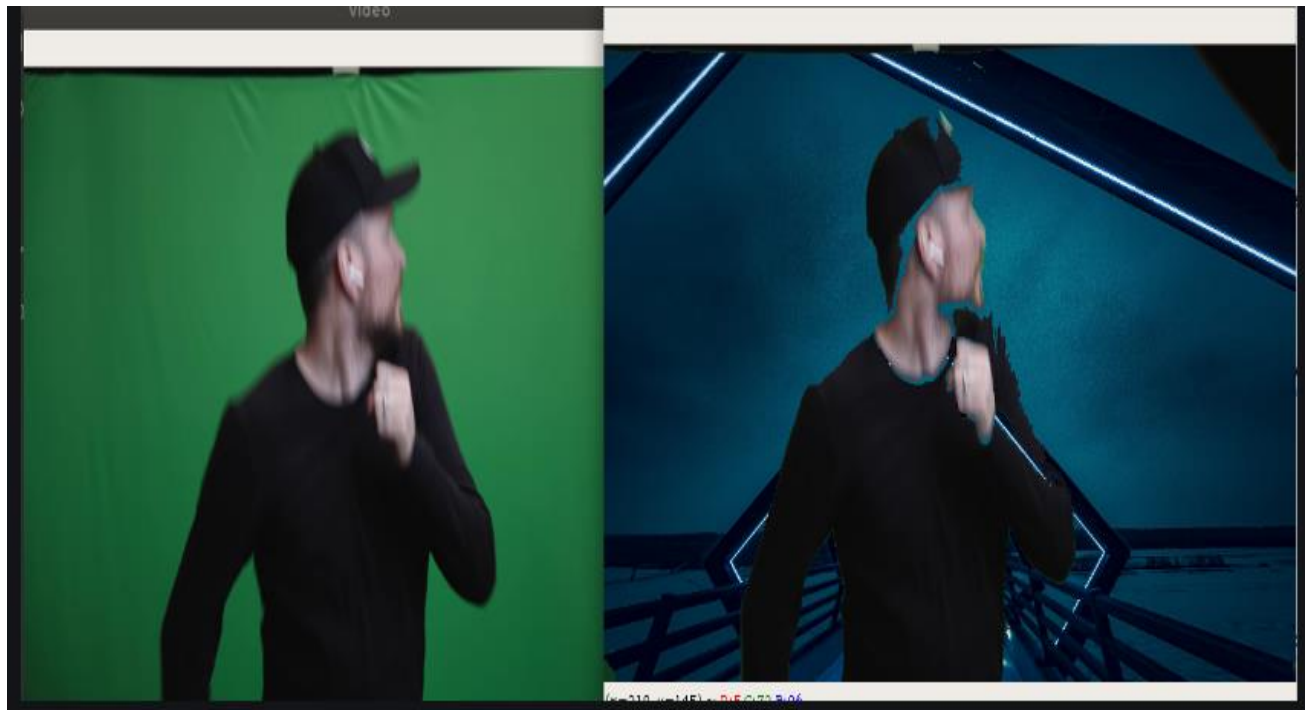
1. Introduction	1
2. Green Screen	3
2.1 Definition	3
2.2 Algorithm	3
2.3 Advantages	4
2.4 Challenges	5
3. Background Problem Statement	6
4. Working of the Project	7
5. System Description	8
6. Project Life Cycle	10
7. System Requirement	11
8. Code	12
9. Conclusion	14
Reference	15

1. Introduction

A green screen is a key component in a film and television production process known as “chroma keying,” in which foregrounded action is combined with separately filmed or constructed background footage. The process works by a person in front of bright green backdrops, then isolating and removing that green colour range, and replacing it with a different background. So, the green screen is a large green backdrop placed in the background of a shot to allow for digital effects later.

Our Green Screen Background Remover System removes the green screen from the original image and adds a static image as a background to the uploaded image and video using OpenCV. It allows for exciting and otherwise-impossible scenarios to be produced, expanding the possibilities.

In this system, the user will need to register first to log in to the system. The user can upload an image with the green screen and the image that they want as a background, then the system will remove the green screen background. As a result, the image will then be displayed to the user with the uploaded background image.



The user can also upload a video with a green screen background and the image for the background. Only static images can be set as a background images. The results will then be displayed to the user with the uploaded background image in a video. The front-end involves Html, CSS, and JavaScript and the back-end involves Python. The framework used is Django and the database is MySQL. Here, we have implemented the OpenCV library.

We live in the era of video calls. Conducted over the internet and using whatever camera that comes with your laptop or computer, we broadcast our lives to our classmates, coworkers, and families. Sometimes, though, we don't want to broadcast our space. My office, like many others, has a few perennial pieces of clutter. I also have a guitar on the wall behind me, which doesn't always scream professionalism. As a result, Zoom and other video calling software includes a feature to hide your background, usually behind an image of your choice. While most don't think of it much, the actual task of determining what determines the foreground and the background in an image is hardly trivial.



It may seem like convincing green screen effects are out of reach for the average video producer, but in fact, with some knowledge, time and preparation, anyone can create believable effects using this technique. But before you get started on your next epic masterpiece, let's take a look at how green screening actually works. Green screen is industry shorthand for a process called chroma keying. This process is used to composite two images together based on color hues. Chroma keying lets you remove a specific color from your image to allow another image to show through from behind. This background image is called a plate, and the more your keyed footage blends in with this plate, the more believable your effects will be.

2. Green Screen

2.1 Definition

A green screen is a large green backdrop placed in the background of a shot to allow for digital effects later. In Post-Production, the special effects department will edit the background from green to... anything they want with CGI (computer-generated imagery) or background footage called "plates." The screen can take up the entire background or just a portion of it, depending on what needs to be replaced.

2.2 Algorithm

As previously mentioned, the pre-packaged background removers in OpenCV will not be used. Instead, the below flow-chart outlines the method I'll use:

First, we'll take the image and convert it to black and white. Next, edge detection will be applied and the contours in the image will be found. Any contours that are too either too big or too small to be the foreground will be removed.

The remaining contours will be considered the foreground. This makes some intuitive sense, as especially small details in a busy background will generate very small contours. By contrast, very large contours which take up most the screen probably aren't the foreground, but some visual artefact of the background.

Finally a mask is generated from the remaining contours and is blended into the original image

2.3 Advantages

Using green screen, or chroma key technology, is beneficial in many ways. Using a green screen allows the filmmaker to layer any background with the subject being filmed or photographed. The subject is filmed against the green (sometimes blue) background, which is replaced in post-production by a digitally generated background.

Besides the most palpable advantage to using a green screen, that is, the ability to **produce essentially any setting** in a controlled environment (green screen studio), there are a number of additional advantages to this technique.



First, shooting on a green screen instead of on-location **saves time**. Because the background added is digitally generated, there is no need to travel beyond the studio to achieve a desired setting. In addition to saving time, this is almost much more **cost effective**.

Filming and photographing in a studio instead of on-location ensures that the shoot will not be **delayed** for reasons beyond the filmmakers' control, such as weather conditions, availability of certain locations, etc. The **controlled environment** provided by a green screen studio ensures that external noise; lighting conditions, etc. do not have to be factored into the filming process.

2.4 Challenges

The green screen is not new to video production and has been a tool used for decades. It allows digital artists to alter the background of the video, using virtually any image that is desired. The options that are available with a green screen are endless.

While a green screen provides many benefits, it can be challenging to work with, if you don't know what you are doing. The biggest challenges that people face with the green screen are obtaining even lighting and avoiding shadows.

Ideally, you want to have as narrow a color range as possible. Even the material you use to create the green screen can impact the lighting. For example, a shiny surface will have areas that reflect the light, while a matte surface will diffuse reflected light and produce a more even color range.

When shadows are created, they can appear as a darker color to the camera and their placement may not be registered. You often see this problem in low-budget films.

3. Background Problem Statement

A green screen is a key component in a film and television production process known as “chroma keying,” in which foregrounded action is combined with separately filmed or constructed background footage.



The process works by a person in front of bright green backdrops, then isolating and removing that green color range, and replacing it with a different background. So, the green screen is a large green backdrop placed in the background of a shot to allow for digital effects later.

Our Green Screen Background Remover System removes the green screen from the original image and adds a static image as a background to the uploaded image and video using OpenCV. It allows for exciting and otherwise-impossible scenarios to be produced, expanding the possibilities.

4. Working of the Project

In this system, the user will need to register first to log in to the system. The user can upload an image with the green screen and the image that they want as a background, then the system will remove the green screen background. As a result, the image will then be displayed to the user with the uploaded background image.

The user can also upload a video with a green screen background and the image for the background. Only static images can be set as a background images. The results will then be displayed to the user with the uploaded background image in a video.



The front-end involves Html, CSS, and JavaScript and the back-end involves Python. The framework used is Django and the database is MySQL. Here, we have implemented the OpenCV library.

5. System Description

The system comprises 1 major module with their sub-modules as follows:

❖ USER:

• Registration

- The user will need to register first to log in to the system.
- The details such as name, age, gender, address, phone number, and Email-ID.

• Login

- The user can log in using their login credentials.

• Upload Image

○ Green Screen Image

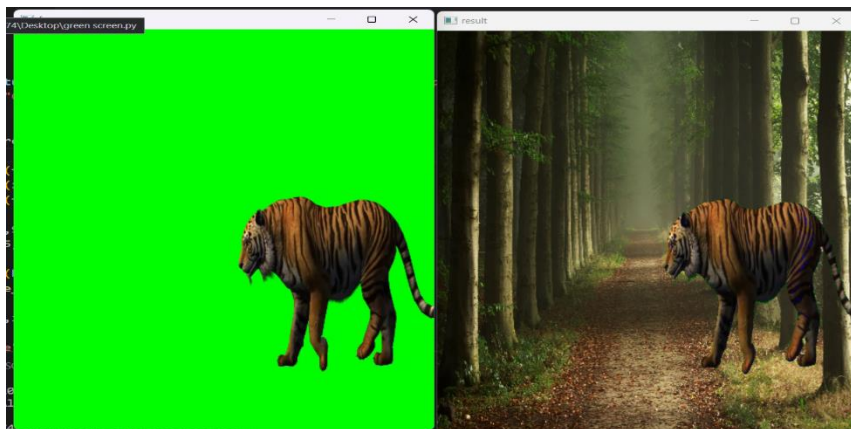
- The user can upload an image with the green screen.

○ Background Image

- Also, need to upload the image that the user wants as a background image.

○ Image Result

- The image result will be displayed to the user.



- **Upload Video**

- **Green screen video**

- The user can upload a video with a green screen background.

- **Background Image**

- They will also need to upload an image for a background.



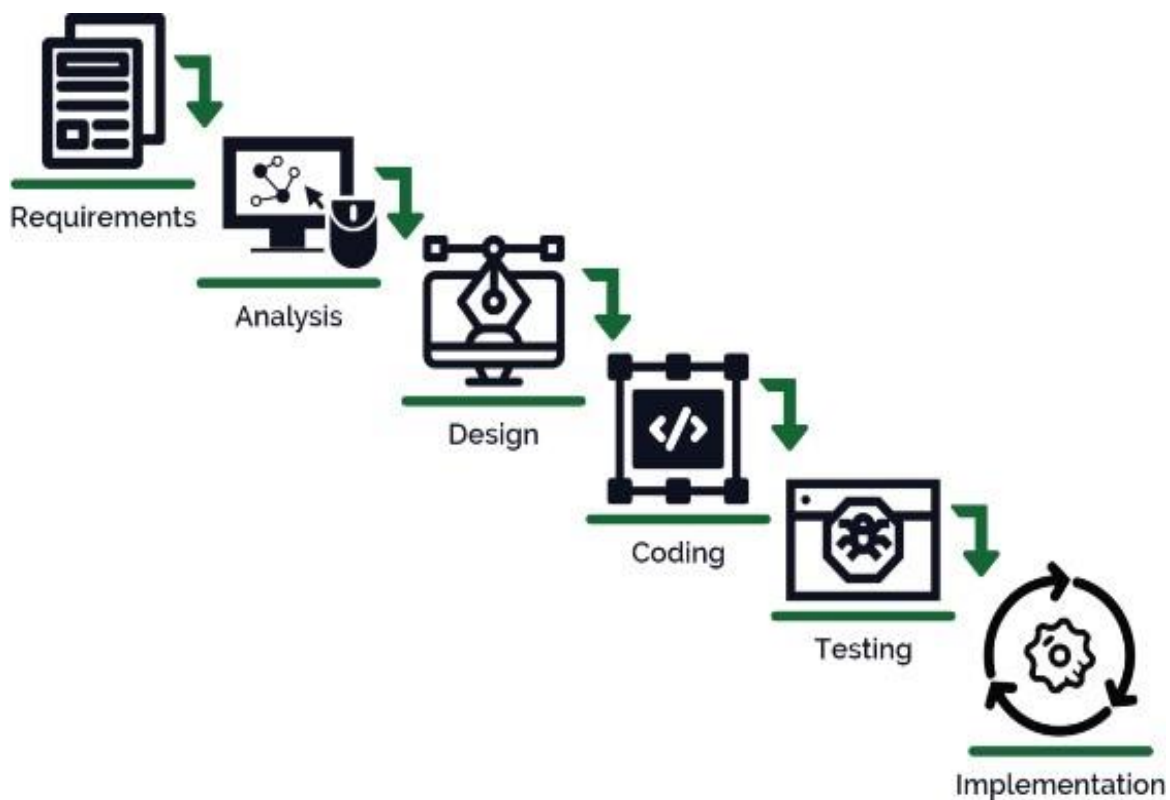
- **Image Result**

- The image result will be displayed to the user.

Buy and pay: Customers who buy through the app can pay through many different payment gateways

6.Project LifeCycle

The waterfall model is a classical model used in the system development life cycle to create a system with a linear and sequential approach. It is termed a waterfall because the model develops systematically from one phase to another in a downward fashion. The waterfall approach does not define the process to go back to the previous phase to handle changes in requirements. The waterfall approach is the earliest approach that was used for software development..



7. System Requirement

I. Hardware Requirement

i.Laptop or PC

- Windows 7 or higher
- I3 processor system or higher
- 4 GB RAM or higher
- 100 GB ROM or higher



II. Software Requirement

ii.Laptop or PC

- Python
- Sublime Text Editor
- XAMP Server

8. Code

1. Import all the necessary libraries.

```
import cv2
import numpy as np
```

2. Load the images or videos.

```
video = cv2.VideoCapture("green.mp4")
image = cv2.imread("bg.jpeg")
```

3. Resize the images and videos to the same size.

```
frame = cv2.resize(frame, (640, 480))
image = cv2.resize(image, (640, 480))
```

4. Load the upper and lower BGR values of green colour.

```
u_green = np.array([104, 153, 70])
l_green = np.array([30, 30, 0])
```

5. Apply the mask and then use bitwise_and

```
mask = cv2.inRange(frame, l_green, u_green)
res = cv2.bitwise_and(frame, frame, mask = mask)
```

6. Subtract the bitwise_and from the original green screen image.

```
f = frame - res  
f = np.where(f == 0, image, f)
```

7. Check for matrix value 0 after subtraction and replace it by the second image.

```
cv2.imshow("video", frame)  
cv2.imshow("mask", f)
```

8. You get the desired result.

```
video.release()  
cv2.destroyAllWindows()
```

9. Conclusion

The intertwined concepts of foreground detection and background subtraction are among some of the most studied aspects of computer vision. While many methods exist, a simple application of edge detection and finding contours within an image provides a good basis.

Using OpenCV's built-in functions, the approach used was able to render background removal in real-time. Under ideal conditions, the algorithm worked near flawlessly, but some additional tweaking may be needed for complex or busy backgrounds.

References

1. Kamal Acharya. School management system project report. Authorea. August 01, 2024. DOI: <https://doi.org/10.22541/au.172254873.34023165/v1>
2. Kamal Acharya. A CASE STUDY OF CINEMA MANAGEMENT SYSTEM PROJECT. Authorea. August 01, 2024. DOI: <https://doi.org/10.22541/au.172254873.30191075/v1>
3. Kamal Acharya. A CASE STUDY ON ONLINE TICKET BOOKING SYSTEM PROJECT. Authorea. August 01, 2024. DOI: <https://doi.org/10.22541/au.172254872.26972790/v1>
4. Kamal Acharya. Web chatting application project report management system. Authorea. August 01, 2024. DOI: <https://doi.org/10.22541/au.172254871.18588592/v1>
5. Kamal Acharya. RETAIL STORE MANAGEMENT SYSTEM PROJECT REPORT. Authorea. August 01, 2024. DOI: <https://doi.org/10.22541/au.172254871.14590154/v1>
6. Kamal Acharya. SUPERMARKET MANAGEMENT SYSTEM PROJECT REPORT. Authorea. August 01, 2024. DOI: <https://doi.org/10.22541/au.172252491.19145062/v1>
7. Kamal Acharya. SOCIAL MEDIA MANAGEMENT SYSTEM PROJECT REPORT. Authorea. August 01, 2024. DOI: <https://doi.org/10.22541/au.172252491.11210579/v1>
8. Kamal Acharya. Online music portal management system project report. Authorea. August 01, 2024. DOI: <https://doi.org/10.22541/au.172252488.89734698/v1>
9. Kamal Acharya. COLLEGE BUS MANAGEMENT SYSTEM PROJECT REPORT. Authorea. July 31, 2024. DOI: <https://doi.org/10.22541/au.172245277.70798942/v1>
10. Kamal Acharya. AUTOMOBILE MANAGEMENT SYSTEM PROJECT REPORT. Authorea. July 31, 2024. DOI: <https://doi.org/10.22541/au.172245276.67982593/v1>
11. Kamal Acharya. Ludo management system project report. Authorea. July 31, 2024. DOI: <https://doi.org/10.22541/au.172243999.98091616/v1>
12. Kamal Acharya. Literature online quiz system project report. Authorea. July 31, 2024 DOI: <https://doi.org/10.22541/au.172243825.53562953/v1>
13. Kamal Acharya. Avoid waste management system project. Authorea. July 29, 2024. DOI: <https://doi.org/10.22541/au.172228528.85022205/v1>
14. Kamal Acharya. CHAT APPLICATION THROUGH CLIENT SERVER MANAGEMENT SYSTEM PROJECT. Authorea. July 29, 2024. DOI: <https://doi.org/10.22541/au.172228527.74316529/v1>
15. Acharya, Kamal, Online Job Portal Management System (May 5, 2024). Available at SSRN: <https://ssrn.com/abstract=4817534> or <http://dx.doi.org/10.2139/ssrn.4817534>
16. Acharya, Kamal, Employee leave management system. (May 7, 2024). Available at SSRN: <https://ssrn.com/abstract=4819626> or <http://dx.doi.org/10.2139/ssrn.4819626>
17. Acharya, Kamal, Online electricity billing project report. (May 7, 2024). Available at SSRN: <https://ssrn.com/abstract=4819630> or <http://dx.doi.org/10.2139/ssrn.4819630>

18. Acharya, Kamal, *POLICY MANAGEMENT SYSTEM PROJECT REPORT*. (December 10, 2023). Available at SSRN: <https://ssrn.com/abstract=4831694> or <http://dx.doi.org/10.2139/ssrn.4831694>
19. Acharya, Kamal, *Online job placement system project report*. (January 10, 2023). Available at SSRN: <https://ssrn.com/abstract=4831638> or <http://dx.doi.org/10.2139/ssrn.4831638>
20. Acharya, Kamal, *Software testing for project report*. (May 16, 2023). Available at SSRN: <https://ssrn.com/abstract=4831028> or <http://dx.doi.org/10.2139/ssrn.4831028>
21. Acharya, Kamal, *ONLINE CRIME REPORTING SYSTEM PROJECT*. (August 10, 2022). Available at SSRN: <https://ssrn.com/abstract=4831015> or <http://dx.doi.org/10.2139/ssrn.4831015>
22. Acharya, Kamal, *Burger ordering system project report*. (October 10, 2022). Available at SSRN: <https://ssrn.com/abstract=4832704> or <http://dx.doi.org/10.2139/ssrn.4832704>
23. Acharya, Kamal, *Teachers Record Management System Project Report* (December 10, 2023). Available at SSRN: <https://ssrn.com/abstract=4833821> or <http://dx.doi.org/10.2139/ssrn.4833821>
24. Acharya, Kamal, *Dairy Management System Project Report* (December 20, 2020). Available at SSRN: <https://ssrn.com/abstract=4835231> or <http://dx.doi.org/10.2139/ssrn.4835231>
25. Acharya, Kamal, *Electrical Shop Management System Project* (December 10, 2019). Available at SSRN: <https://ssrn.com/abstract=4835238> or <http://dx.doi.org/10.2139/ssrn.4835238>
26. Acharya, Kamal, *Online book store management system project report*. (February 10, 2020). Available at SSRN: <https://ssrn.com/abstract=4835277> or <http://dx.doi.org/10.2139/ssrn.4835277>
27. Acharya, Kamal, *Paint shop management system project report*. (January 10, 2019). Available at SSRN: <https://ssrn.com/abstract=4835441> or <http://dx.doi.org/10.2139/ssrn.4835441>
28. Acharya, Kamal, *Supermarket billing system project report*. (August 10, 2021). Available at SSRN: <https://ssrn.com/abstract=4835474> or <http://dx.doi.org/10.2139/ssrn.4835474>
29. Acharya, Kamal, *Online taxi booking system project report*. (March 10, 2022). Available at SSRN: <https://ssrn.com/abstract=4837729> or <http://dx.doi.org/10.2139/ssrn.4837729>
30. Acharya, Kamal, *Online car servicing system project report*. (March 10, 2023). Available at SSRN: <https://ssrn.com/abstract=4837832> or <http://dx.doi.org/10.2139/ssrn.4837832>
31. Acharya, Kamal, *School management system project report*. (July 10, 2021). Available at SSRN: <https://ssrn.com/abstract=4837837> or <http://dx.doi.org/10.2139/ssrn.4837837>
32. Acharya, Kamal, *Furniture Showroom Management System Project Report* (March 21, 2021). Available at SSRN: <https://ssrn.com/abstract=4839422> or <http://dx.doi.org/10.2139/ssrn.4839422>
33. Acharya, Kamal, *Online Vehicle Rental System Project Report* (March 21, 2019). Available at SSRN: <https://ssrn.com/abstract=4839429> or <http://dx.doi.org/10.2139/ssrn.4839429>
34. Acharya, Kamal, *Fruit Shop Management System Project Report* (August 10, 2023). Available at SSRN: <https://ssrn.com/abstract=4841048> or <http://dx.doi.org/10.2139/ssrn.4841048>

35. Acharya, Kamal, *Hall Booking Management System Project Report* (December 21, 2023). Available at SSRN: <https://ssrn.com/abstract=4841055> or <http://dx.doi.org/10.2139/ssrn.4841055>
36. Acharya, Kamal, *Lundry Management System Project Report* (October 21, 2023). Available at SSRN: <https://ssrn.com/abstract=4841059> or <http://dx.doi.org/10.2139/ssrn.4841059>
37. Acharya, Kamal, *A CASE STUDY OF CINEMA MANAGEMENT SYSTEM PROJECT* (September 25, 2023). Available at SSRN: <https://ssrn.com/abstract=4841209> or <http://dx.doi.org/10.2139/ssrn.4841209>
38. Acharya, Kamal, *A CASE STUDY ON ONLINE TICKET BOOKING SYSTEM PROJECT* (May 25, 2024). Available at SSRN: <https://ssrn.com/abstract=4841210> or <http://dx.doi.org/10.2139/ssrn.4841210>
39. Acharya, Kamal, *ONLINE DATING MANAGEMENT SYSTEM PROJECT REPORT*. (April 25, 2023). Available at SSRN: <https://ssrn.com/abstract=4842066> or <http://dx.doi.org/10.2139/ssrn.4842066>
40. Acharya, Kamal, *ONLINE RESUME BUILDER MANAGEMENT SYSTEM PROJECT REPORT*. (April 25, 2021). Available at SSRN: <https://ssrn.com/abstract=4842071> or <http://dx.doi.org/10.2139/ssrn.4842071>
41. Acharya, Kamal, *TOLL TEX MANAGEMENT SYSTEM PROJECT REPORT* (August 21, 2023). Available at SSRN: <https://ssrn.com/abstract=4842082> or <http://dx.doi.org/10.2139/ssrn.4842082>
42. Acharya, Kamal, *Chat Application Through Client Server Management System Project Report* (June 25, 2023). Available at SSRN: <https://ssrn.com/abstract=4842761> or <http://dx.doi.org/10.2139/ssrn.4842761>
43. Acharya, Kamal, *Web Chatting Application Management System Project Report* (April 25, 2022). Available at SSRN: <https://ssrn.com/abstract=4842771> or <http://dx.doi.org/10.2139/ssrn.4842771>
44. Acharya, Kamal, *Automobile management system project report* (May 25, 2022). Available at SSRN: <https://ssrn.com/abstract=4846917> or <http://dx.doi.org/10.2139/ssrn.4846917>
45. Acharya, Kamal, *College bus management system project report* (April 25, 2023). Available at SSRN: <https://ssrn.com/abstract=4846920> or <http://dx.doi.org/10.2139/ssrn.4846920>
46. Acharya, Kamal, *Courier management system project report* (May 25, 2023). Available at SSRN: <https://ssrn.com/abstract=4846922> or <http://dx.doi.org/10.2139/ssrn.4846922>
47. Acharya, Kamal, *Event management system project report* (April 25, 2021). Available at SSRN: <https://ssrn.com/abstract=4846927> or <http://dx.doi.org/10.2139/ssrn.4846927>
48. Acharya, Kamal, *Library management system project report II* (May 25, 2020). Available at SSRN: <https://ssrn.com/abstract=4848857> or <http://dx.doi.org/10.2139/ssrn.4848857>